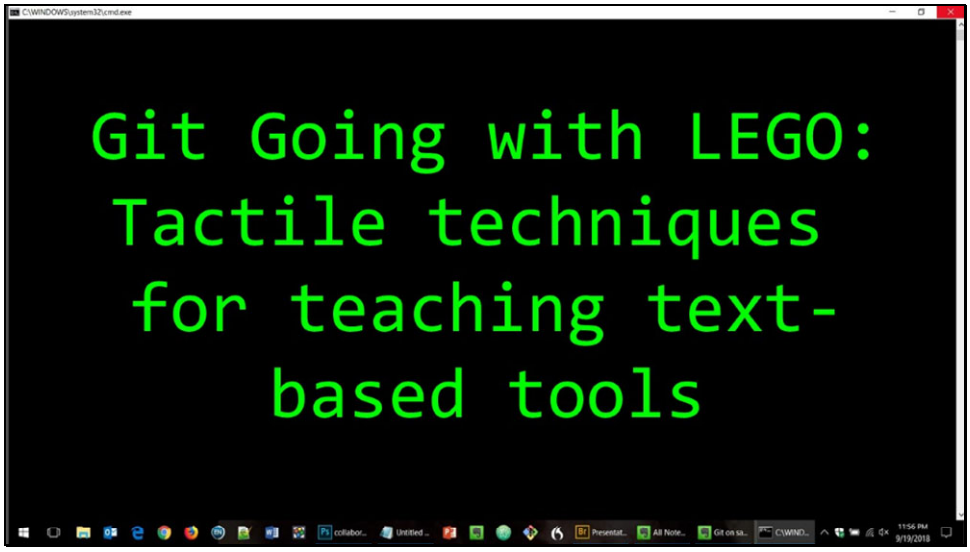
Slide 1

Our joke throughout the early stages of what we're presenting is that this is what happened when a Software Carpentry Instructor and a LEGO Serious Play Facilitator walked into a bar.

Slide 2



FORCE2018 Conference; October 10-12, 2018; Montreal, QC, Canada

Speaker: Tami

Welcome to our presentation Git Going with Lego: Tactile techniques for teaching text-based tools.

Slide 3

Speaker: Tami

We are:

Jamene Brooks-Kieffer
Software Carpentry instructor
Assoc. Librarian / Data Services

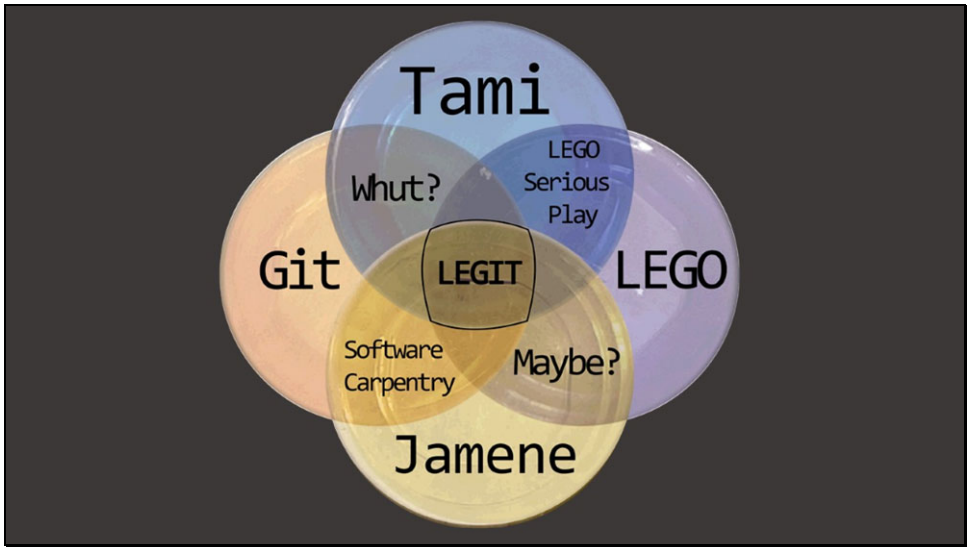
And

Tami Albin
LEGO Serious Play facilitator
Assoc. Librarian / Faculty/Staff Initiatives & Engagement

And we are from the University of Kansas in Lawrence, Kansas.

Please keep all Wizard of Oz jokes to yourselves. We've heard them all.

Slide 4



Speaker: Tami

We are librarians; this is our obligatory Venn diagram.

So how did this LEGit collaboration come into existence?

Tami's thoughts:

- Our cubicles are across from one another.
- Tami has been trained by Jamene in data management basics
- Tami's office is full of toys and is interested in serious play
- Tami was trained as a Lego Serious Play facilitator
- Tami is up for learning new things

Jamene's thoughts:
After teaching Git in standalone workshops and as part of Software Carpentry workshops, Jamene was looking for a way to help novice learners understand the basics of Git a little better. Tami had returned from LEGO Serious Play Facilitator training and was beginning to accumulate LEGO kits for other workshops.
Jamene wondered if there would be a way to work building with LEGO bricks into the Software Carpentry Git lesson.

In starting this collaboration we have found that we don't have to be the same, do the same jobs, have the same knowledge or skills to work together successfully.

Slide 5

Speaker: Jamene

So now that you know a bit about us, we'd like to get a sense of the folks in the room. You have received a yellow and a blue sticky note. Blue is yes; yellow is no.

Use your sticky notes to respond:
Indicate your experience with Git – blue for yes (any experience at all) and yellow for no, none.
Indicate your experience with LEGO – again, blue for yes and yellow for no
Indicate whether you have had any contact with the Carpentries foundation, perhaps through a Software or Data Carpentry workshop or through the Carpentries Foundation – blue for yes; yellow for no

You've just gotten a taste of a Carpentries-style assessment of the level of knowledge in the room about a particular topic.

Slide 6

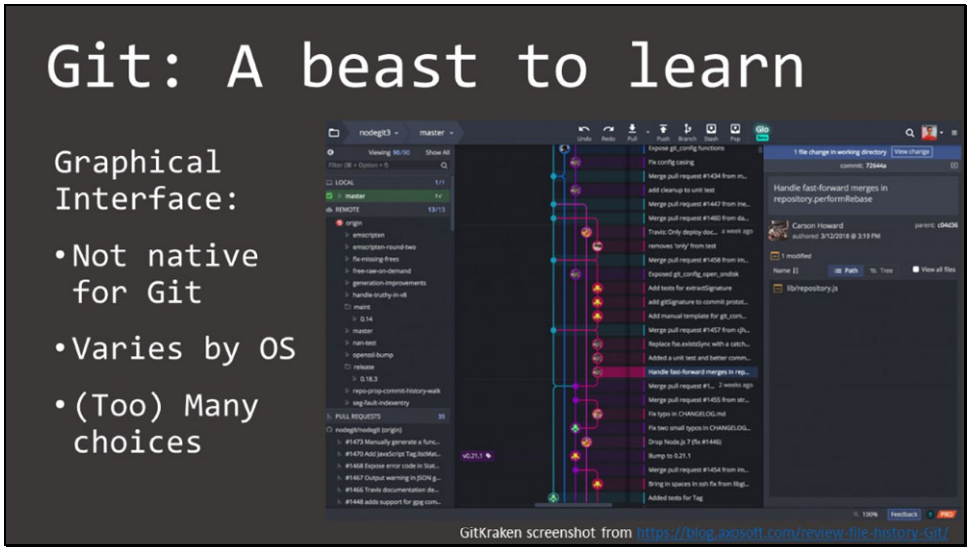
Speaker: Jamene

In this presentation we are proposing that Git is a tool of Scholarly Communication.

Git is first and foremost an open source version control tool that allows us to record the history of changes to our project without maintaining multiple files that all have slightly different names. Git records details like the day and time changes were made and who made the changes, as well as what the changes were. Git was designed as a version control tool for software development, but researchers in many disciplines use it to track changes to projects that don't necessarily involve code.

Git can enhance collaboration and sharing primarily through cloud services that run Git's software (e.g.: GitHub, GitLab, Bitbucket) and allow multiple collaborators to contribute to a project without having to email files back and forth. And because Git maintains a complete record of changes to a project, it enhances reproducibility and transparency.

Slide 7



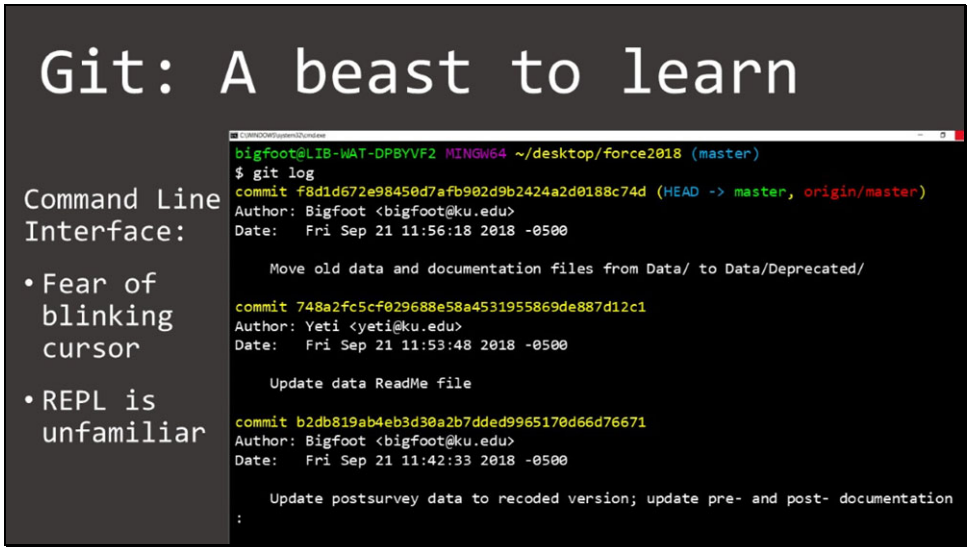
Speaker: Jamene

There is no one graphical interface for Git because Git was not created for the GUI (Graphical User Interface). Instead, there are multiple desktop clients that visualize Git workflows in a GUI that's often more familiar to new users.
This screenshot is of a popular Git GUI called GitKraken.

I would argue from a novice learner perspective that this is hard enough to learn. But it's really hard to teach Git in a GUI because there is no one GUI to rule them all, so as the instructor you don't know exactly what the learner is seeing. It's also easy to get wrapped up in what the GUI client can do and forget about teaching what Git is doing – which is the point.

So robust lessons such as those offered by the Carpentries teach…

Slide 8



Speaker: Jamene

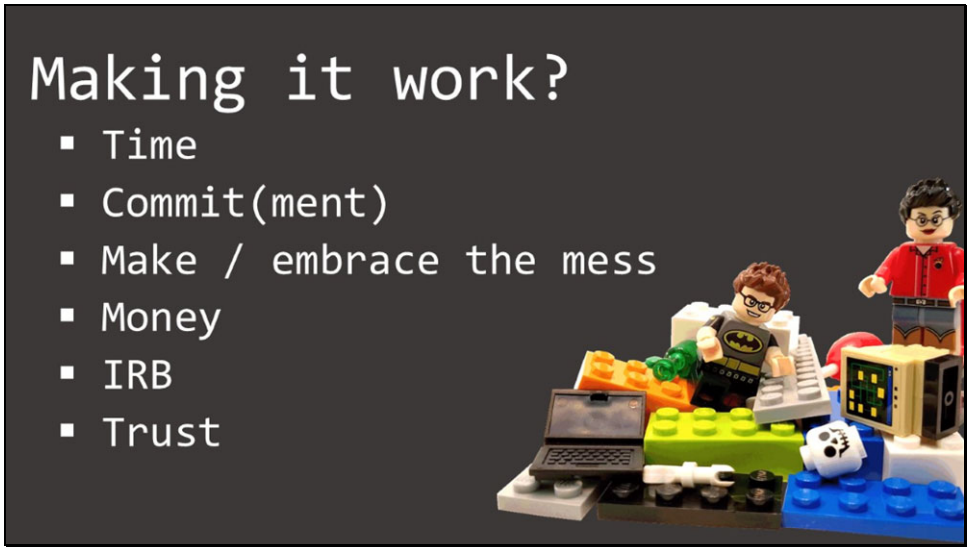
…the Command Line Interface (CLI)
The CLI is native for Git, but comes with its own challenges for novice learners.
The big one is fear of the blank screen and blinking cursor – if you don't know what to type, you can't even get out, much less get work done.

And the way that the CLI gives you feedback on what you're trying to do – called REPL, or Read-Eval-Print Loop – is really strange compared to the point-and-click experience that most of us are used to.

In either interface, Git presents a challenge for the novice learner and the instructor because:
- Its language and processes aren't intuitive
- It is operating at a meta level on the aboutness of files, not on the files themselves, in a way that's invisible unless you know how to look

REPL = read-eval-print loop:
https://en.wikipedia.org/wiki/Read%E2%80%93eval%E2%80%93print_loop

Slide 9



Speaker: Tami

In making this collaboration work, we found some elements that were critical to our approach:

Time:
Investing time in regular meetings talking through how or if these two things could go together – like 2 and 3 hour meetings (that we both looked forward to, fwiw). We preserved our time for these meetings at the expense of other things and didn't let our general busyness get in the way of our collaboration.

Commit:
Commit to what you agree to do.
Commit to learning new things in haphazard ways
Commit to respecting your colleague and do your part
Commit to being honest - when you don't understand, get confused, get frustrated or have a brain melt.
Commit to a healthy process, not just a final project
Commit to vulnerability and sharing what you do well and not so well
Commit to having fun while you learn

Mess:
Yes, we made literal messes of LEGO on our meeting room table, but we also made messes in our brains –
For Tami, this was the mess of learning Git in a one-on-one, not terribly methodical crash course interspersed with LEGO building and a lot of "So, wait…"
For Jamene, this was the mess of learning about LEGO bricks and mapping the process of building a LEGO structure (out of what kind of bricks? How big? How much to include/leave out?) onto the Software Carpentry Git lesson materials.

Funding
Money:
If we're going to teach with LEGO, we need LEGO. So we asked for LEGO money via the Libraries' research funding program. And we got some! Then Tami kept buying LEGO…

IRB:
If we're going to assess the effect of LEGO on learning Git, we need to get the okay to do human subjects research.

Trust
If you don't think that your colleague is on board or have concerns about the project not working out, get out

Slide 10

Speaker: Tami

The process of creating our IRB application was a chance to dig into the available literature on, essentially, teaching computing topics using tangible experiences and serious play as a way to enhance tangible experiences.

We are not going to talk about the theories behind what we're doing (because this presentation is happening at warp speed), other than to mention the two major sources:

- Cognitive Load Theory – basically, using neuroscience to maximize what people can learn, remember, and apply
- Serious Play – is goal oriented, concerned with outcomes and process, and lastly, connected to an identified purpose. By using LEGO, participants engage in constructionism (creating something external to themselves) and concrete thinking (thinking with and through tangible objects)

We have a massive bibliography that you can dig into on your own.

Slide 11

Speaker: Tami

Research:
- Pre- and post-workshop anonymous surveys, distributed on paper
- Invitation for follow-up one-on-one interviews
- Collect/tabulate/analyze surveys
- Schedule follow-up interviews (forthcoming)

Teaching:
- Schedule and organize workshop (Software Carpentry, 2018-08-16)
- Distribute LEGO kits at workshop
- Teach workshop
- Repeat (forthcoming)

Slide 12

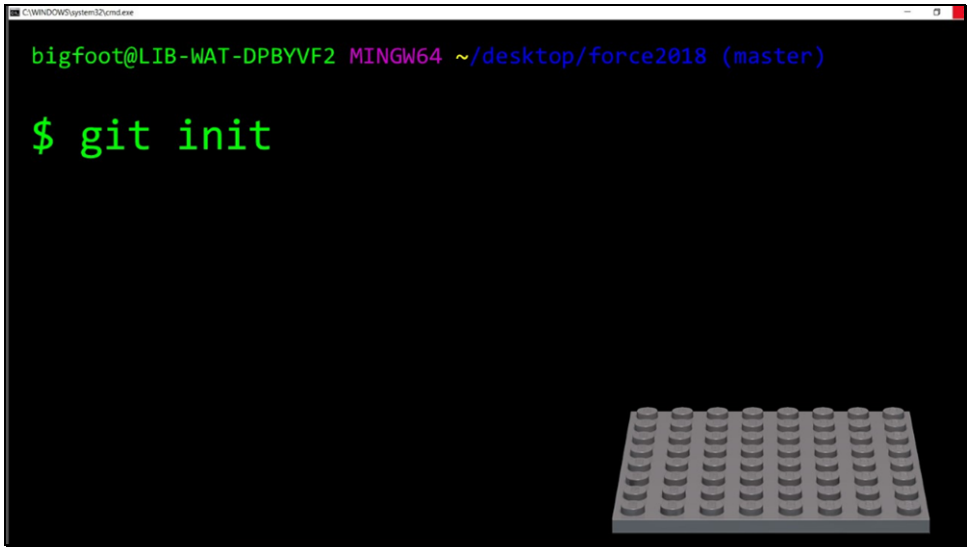
Speaker: Tami

After much debate and discussion we chose the following bricks:

- 8x8 plate represents git init
- 2x4 bricks represent files
- Matching 2x4 plates represent git add
- Black 2x4 plates represent git commit

There is no rhyme or reason to the colour scheme other than these were the bricks they had in stock when I called the Lego company or order the bricks

Slide 13



Speaker: Jamene

This is a representation of the command line screen (also called a shell) that learners are using during the workshop.
The top line is part of the prompt that is automatically generated by the shell. The $ is where you type things.

Learners in the workshop are following instructions to know what to type. While they type the commands, they are also building with the pictured LEGO piece.

The first command that learners use is the one that creates a new git repository: git init
Then they lay down the gray 8x8 plate that represents the beginning of their repository.

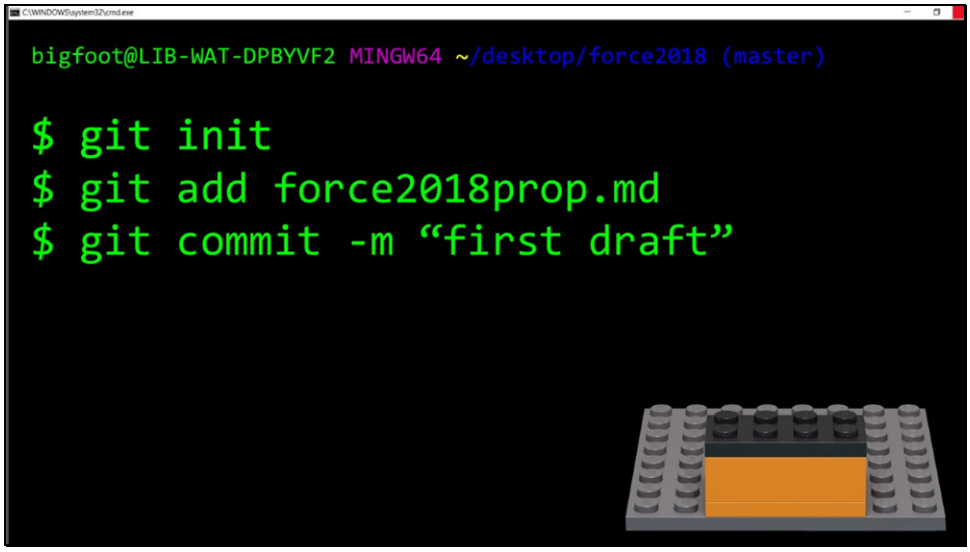
Slide 14

Speaker: Jamene

As part of the workshop, the learners have created a text file to work with in git. The file is represented by the 2x4 brick.

In order to tell git to pay attention to their new file, learners issue a command: git add (plus the name of the file).
This action is represented by placing the 2x4 plate (which represents “git add”) and the matching 2x4 brick (representing the file) onto the 8x8 plate that represents the git repository.

Slide 15



Speaker: Jamene

Getting git to pay attention to the changes made to a file is a two-step process that always trips up learners the first few times.
Issuing the “git add” command is necessary but not sufficient. Learners also have to issue the “git commit” command in order to complete the process of recording the change to the git repository.
The commit is always represented by a black 2x4 plate. When learners type “git commit –m “a commit message to my future self” they also lay a black plate onto the top of their LEGO build.

Slide 16

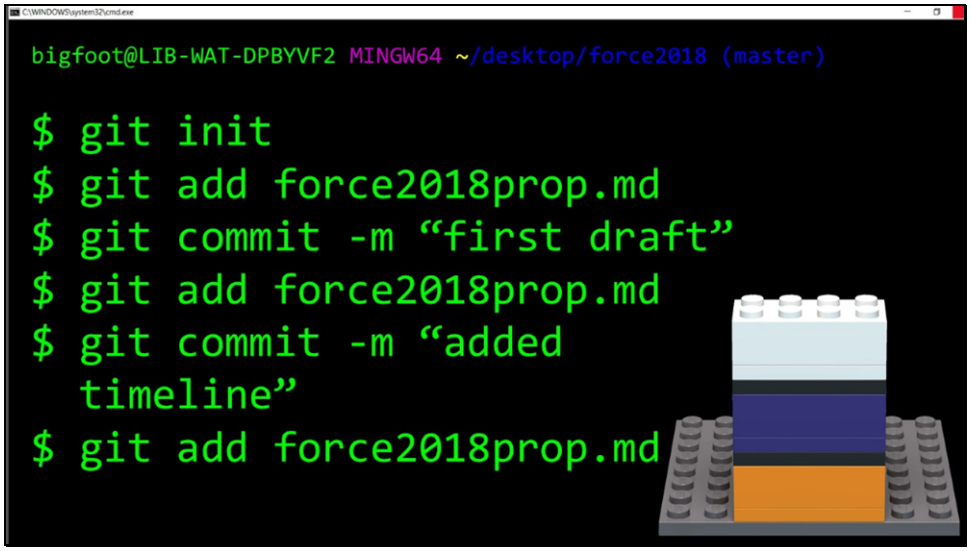
Speaker: Jamene

Throughout the workshop, learners continue to modify their file in different scenarios and add/commit subsequent changes to their git repository.
Repeating the commands is necessary for the process to sink in; it’s also necessary to understanding the workflow that one establishes when using git for version control of files or a project.
Learners complete a total of 5 commits over the course of the workshop. The following slides (17-23) represent how the LEGO build grows as the workshop progresses.

Slide 17

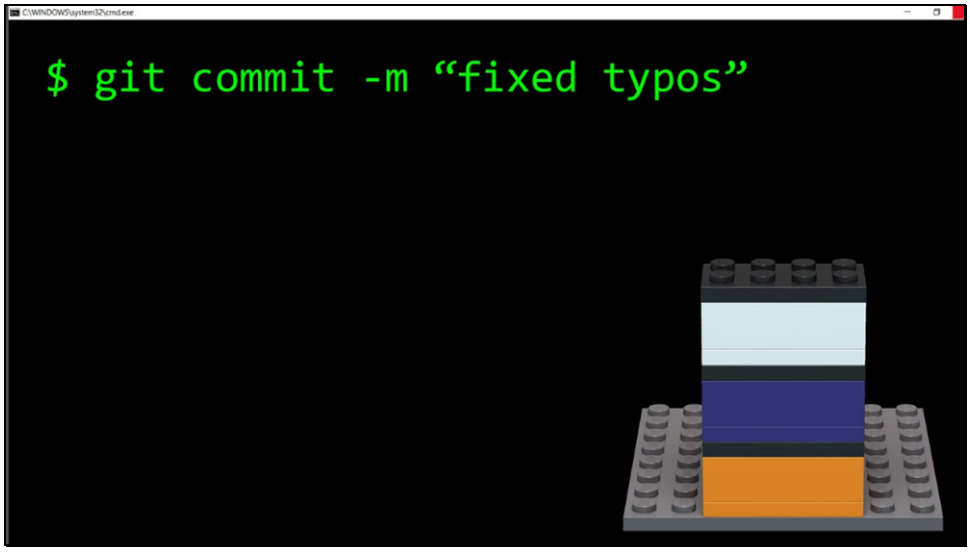
Speaker: Jamene

Slide 18



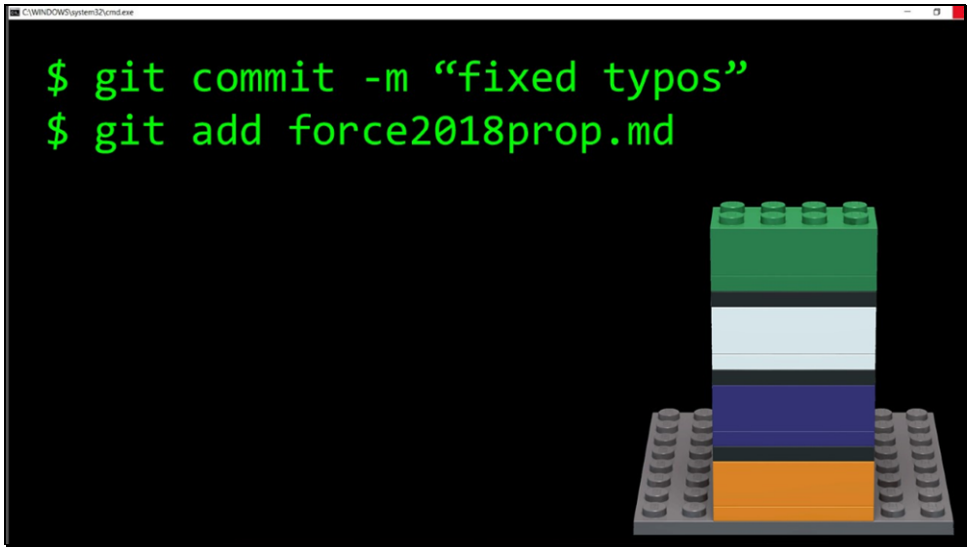
Speaker: Jamene

Slide 19



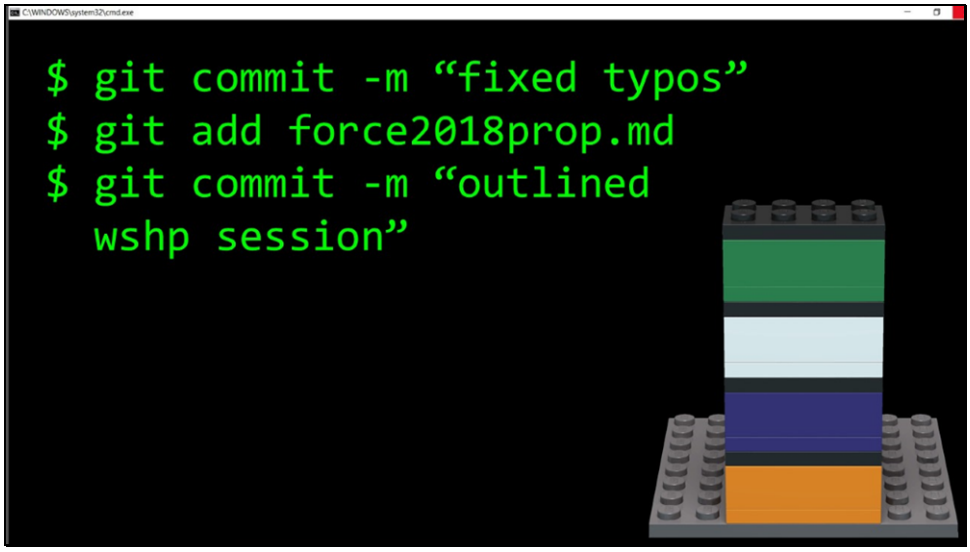
Speaker: Jamene

Slide 20



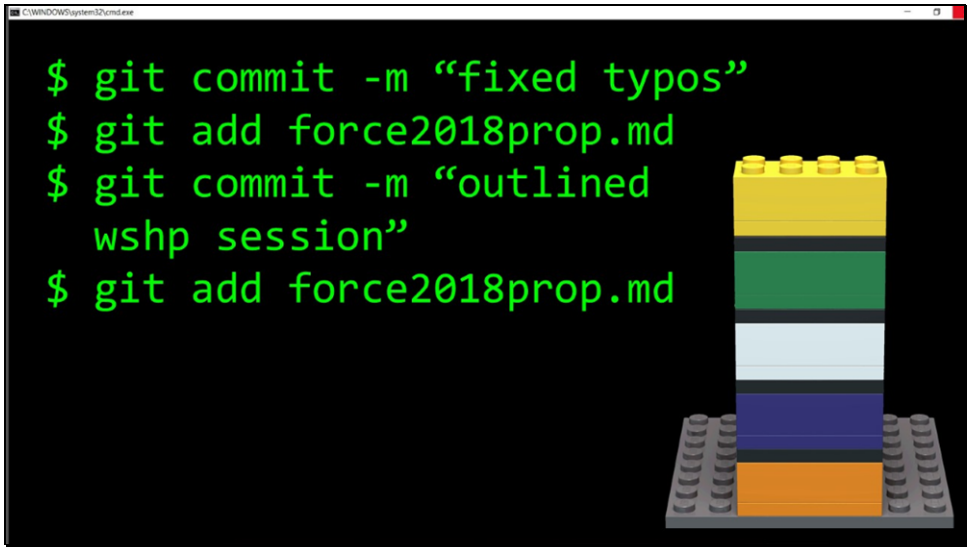
Speaker: Jamene

Slide 21



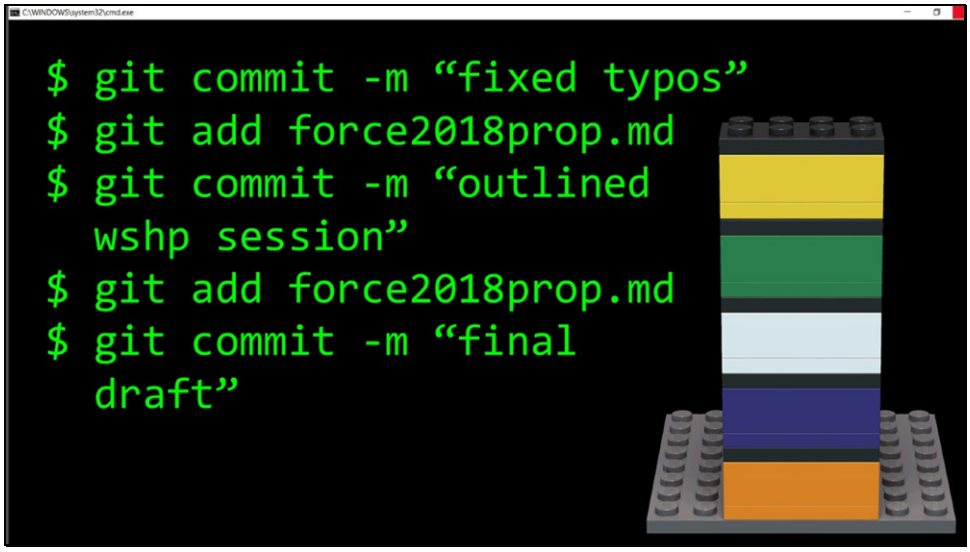
Speaker: Jamene

Slide 22



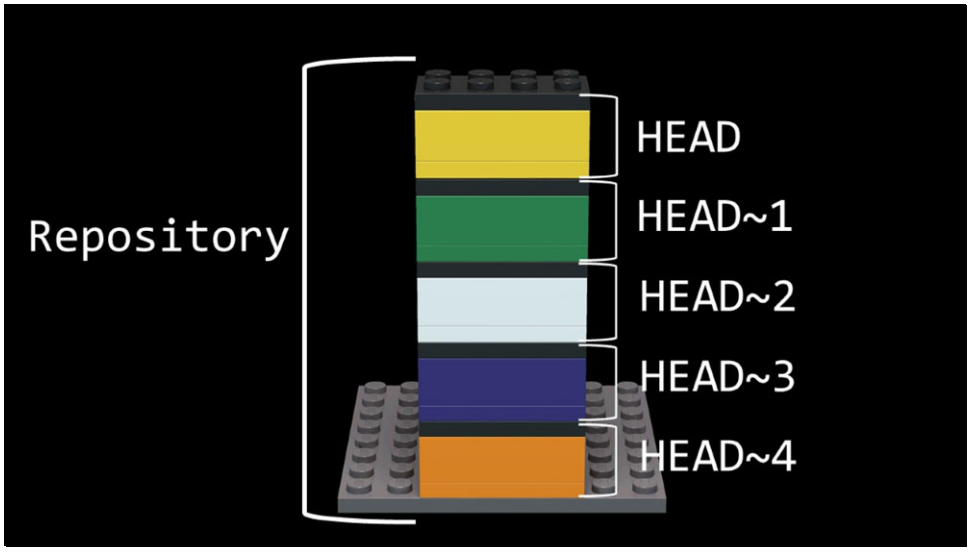
Speaker: Jamene

Slide 23



Speaker: Jamene

Slide 24



Speaker: Jamene

As the LEGO build grows, learners can see a structure that mirrors the history of changes to the file that git is recording in the repository. We use the tower as a visual and tactile reference for talking about exploring the history of the repository using git's HEAD convention.

Slide 25

Speaker: Tami

Where we are at:

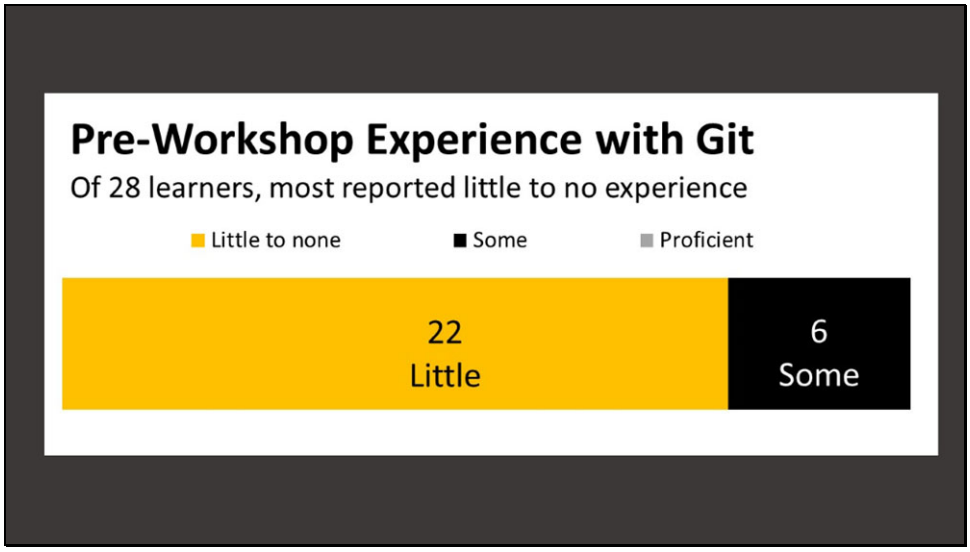
- Tiny data set from one Git workshop – we will show you some of the results in a moment
- Planning for post-Git workshop interviews
- Learning and developing skills with tools such as python, pandas, markdown, GitHub, GitLab, Git - wrestling with doing the work of research using open software and open practices – just as we push the researchers we talk to to do
- Discussions around teaching future Git workshops – one off workshops and/or additional Software Carpentry workshops,, plus the need for colleagues in the Libraries to understand Git for consultations and instruction with faculty, staff, and graduate students on campus, as well as their own scholarly research.

Slide 26

Speaker: Tami

What we have found out so far:

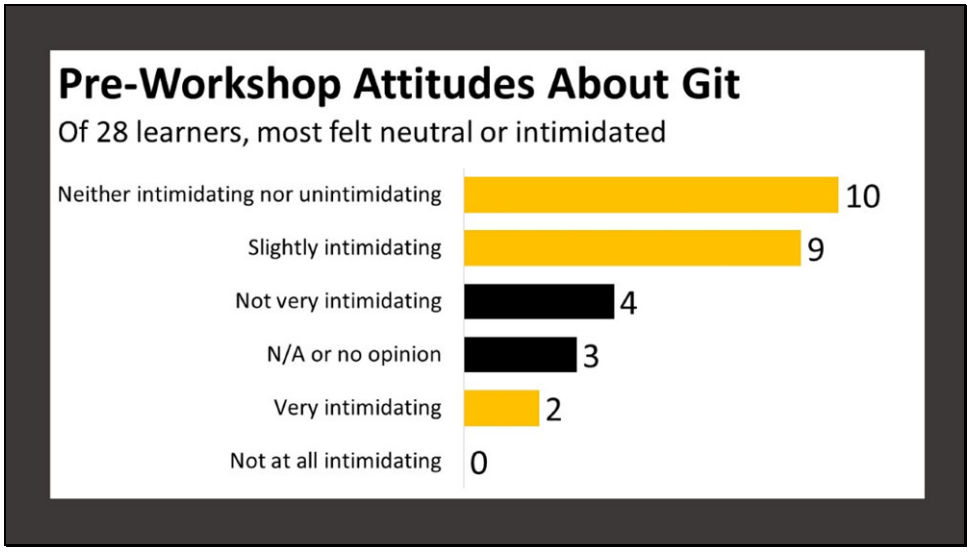
Slide 27



Speaker: Jamene

In the pre-workshop survey, most reported little to no experience with Git. No one indicated that they were proficient.

Slide 28



Speaker: Jamene

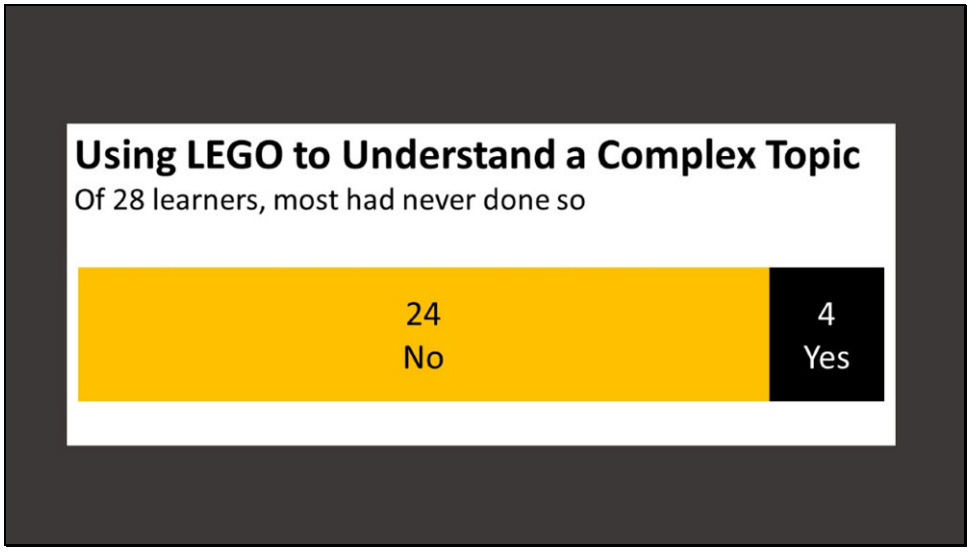
In the pre-workshop survey, most reported that they felt either neutral or intimidated by Git.

Slide 29

Speaker: Jamene

In the pre-workshop survey, most reported that they had never used LEGO in a professional capacity.

Slide 30



Speaker: Jamene

In the pre-workshop survey, most reported that they had never used LEGO to understand a complex topic.

So we were working with a population of learners who were mostly new to Git, somewhat intimidated by Git, and mostly unfamiliar with applying LEGO to problem-solving in a professional setting.

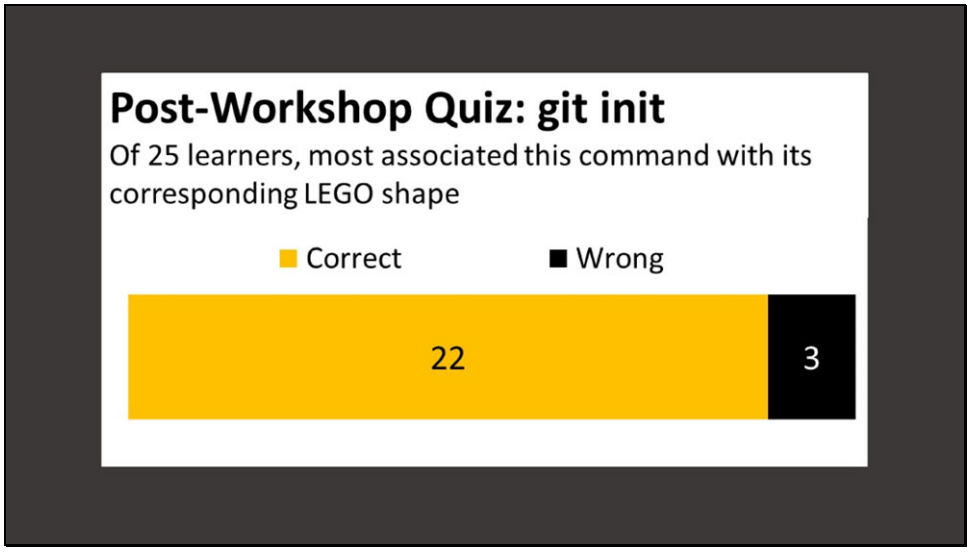
Slide 31

Speaker: Jamene

We had fewer responses to the post-workshop survey because some people had to leave early.

In the post-workshop survey, most reported that the lesson material was new to them – not a surprise given the pre-workshop results.

Slide 32

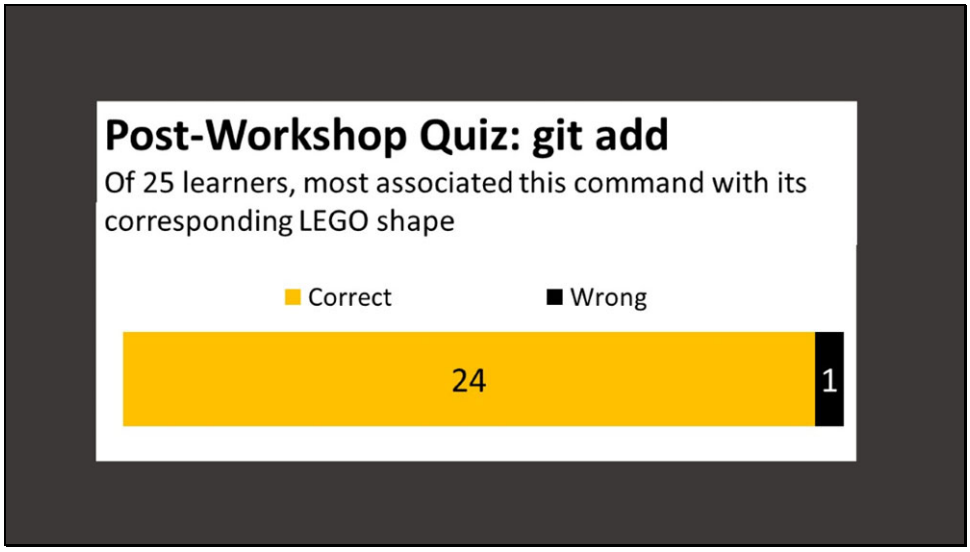


Speaker: Jamene

Part of the post-workshop survey was a quiz where we asked respondents to match up the appropriate git command with the LEGO piece or stage of the build that corresponded to that command.
The three commands we included in the quiz were “git init”, “git add”, and “git commit”

Most respondents correctly matched up git init with its LEGO representation.

Slide 33

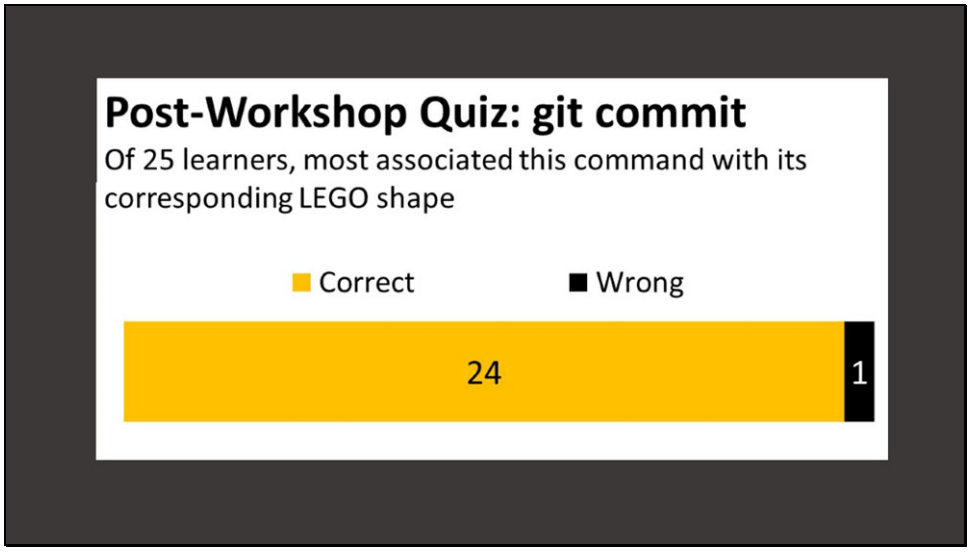


Speaker: Jamene

Part of the post-workshop survey was a quiz where we asked respondents to match up the appropriate git command with the LEGO piece or stage of the build that corresponded to that command.
The three commands we included in the quiz were “git init”, “git add”, and “git commit”

Most respondents correctly matched up git add with its LEGO representation.

Slide 34

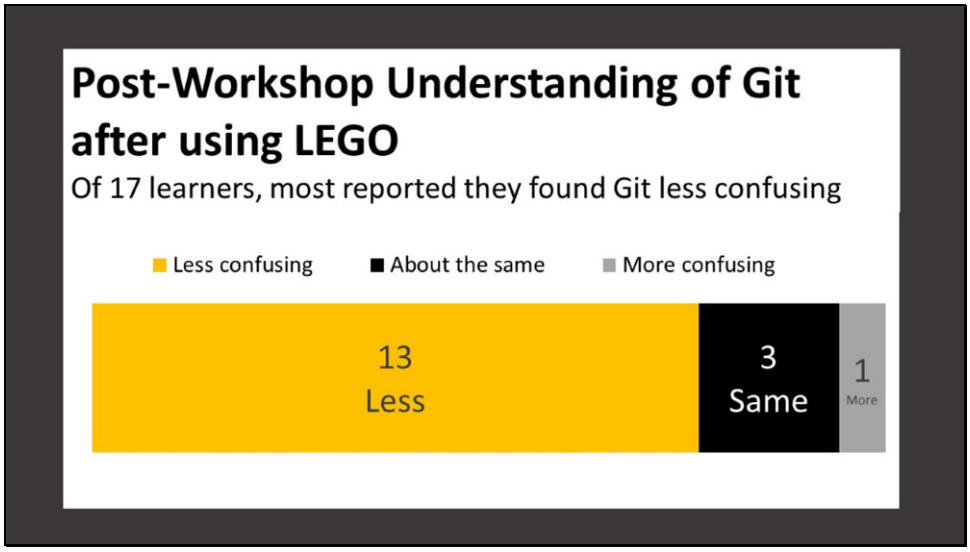


Speaker: Jamene

Part of the post-workshop survey was a quiz where we asked respondents to match up the appropriate git command with the LEGO piece or stage of the build that corresponded to that command.
The three commands we included in the quiz were “git init”, “git add”, and “git commit”

Most respondents correctly matched up git commit with its LEGO representation.

Slide 35

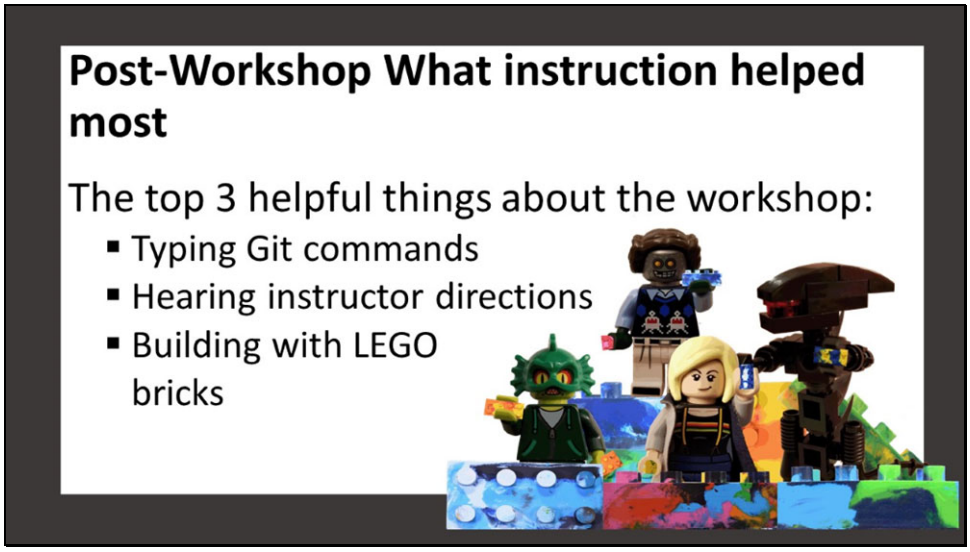


Speaker: Jamene

This portion of the post-workshop survey has even fewer respondents because we had to place this question on the back of the page. Some respondents didn't see or chose to skip this question, so we have fewer responses.

In the post-workshop survey, most respondents indicated that they found Git less confusing after working with LEGO.

Slide 36



Speaker: Jamene

We wound up putting a dreadful question on the post-workshop survey, in the sense that it was difficult for respondents to interpret and answer in the way we had intended, and it was therefore difficult for us to analyze the results. So there is no neat chart for this question. Instead, we can say generally that the three most helpful parts of the workshop that respondents chose from a list of possibilities were:

- Typing Git commands
- Hearing directions from the instructor
- Building with LEGO bricks

The actual text of post-workshop question 2 follows:

2) What part of the instruction on Git helped you the most?
(Please rank in order from 1 - more useful to 6 - less useful.)

_____ Discussing Git with other participants
_____ Hearing other participants' questions
_____ Typing Git commands
_____ Building with LEGO bricks
_____ Hearing directions from instructor
_____ Other: _____

Slide 37

Speaker: Tami

We mentioned earlier where we are presently with

- Interviews
- More workshops
- More Git
- More Python

We are also thinking of ways to use LEGO to discuss more advanced Git functions such as pushing, pulling, branching, cloning. In order to do this we need to expand our knowledge of Git and Tami will definitely need to buy more LEGO.

Slide 38

Thank you so much.